Applicants: Talyansky, Roman et al. Assignee: Intel Corporation
Serial Number: 10/673,261 Attorney Docket: P-6115-US

RECEIVED
CENTRAL FAX CENTER

JAN 0 4 2007

Amendments to the Claims

The following listing of claims replaces all prior versions and listings of claims in the application:

1. (Currently Amended) A method comprising allocating spill cells used by an instrumentation fragment that has access to ~~a single~~ only one free register and that is run on a processor with a register stack architecture.

2. (Currently Amended) ~~A~~ The method ~~as in~~ of claim 1, where said allocating comprises:
designating an index for a spill array and a lock array;
incrementing said index;
loading said incremented index in said free register;
altering a value in a cell of said lock array;
determining whether said altered value in said cell of said lock array equals a pre-defined value; and
allocating said spill cell corresponding to said incremented index.

3. (Currently Amended) ~~A~~ The method ~~as in~~ of claim 2, further comprising reducing said incremented index by the number of cells in said lock array if said incremented index exceeds the number of cells in said lock array.

4. (Currently Amended) ~~A~~ The method ~~as in~~ of claim 2, wherein said incrementing said index comprises executing a threadsafe instruction.

5. (Currently Amended) ~~A~~ The method ~~as in~~ of claim 2, wherein said determining whether said altered value equals a pre-defined value comprises freeing a predicate register.

6. (Currently Amended) ~~A~~ The method ~~as in~~ of claim 2, wherein said altering said value in said lock array comprises incrementing said value.

7. (Currently Amended) ~~A~~ The method ~~as in~~ of claim 2, comprising reducing said altered value of said lock array if said altered value equals a maximum permitted value.

8. (Currently Amended) A method using only one free register of a processor comprising:

storing an incremented index in a the free register of a processor, such processor using a register stack architecture;

calculating in said free register the address of a cell of a first array corresponding to said incremented index, said cell storing a value equal to said index;

loading in said free register an incremented value from said cell of said first array;

comparing said incremented value in said free register to a pre-defined value; and

allocating a cell of a second array corresponding to said index ~~in said free register~~ if said incremented value equals said predefined value.

9. (Currently Amended) ~~A~~ The method ~~as in~~ of claim 8, comprising reducing said incremented index modulo to the number of cells in said first array.

10. (Currently Amended) ~~A~~ The method ~~as in~~ of claim 8, comprising reducing said incremented value if said incremented value equals a maximum permitted value.

11. (Currently Amended) A method of spill cell allocation using only one free register of a processor that uses a register stack architecture comprising:

storing an incremented value in a memory and in a the free register ~~of a processor that uses a register stack architecture~~;

comparing said incremented value in said free register to a pre-defined value;

allocating a spill cell if said incremented value in said free register equals said pre-defined value; and

re-setting said incremented value in said memory.

12. (Currently Amended) ~~A~~ The method ~~as in~~ of claim 11, comprising determining if said incremented value equals a maximum permitted value.

13. (Currently Amended) ~~A~~ The method ~~as in~~ of claim 11, comprising reducing said incremented value if said incremented value equals a maximum permitted value.

14. (Currently Amended) A device comprising a processor with a register stack architecture, said device capable of allocating a spill cell using only one free register.

15. (Currently Amended) ~~A~~ The device ~~as in~~ of claim 14, said processor to:

store an incremented index of an array in said free register;

calculate in said free register the address of a cell of an array corresponding to said incremented index, said cell storing a value equal to said index;

load in said free register an incremented value from said cell of said array;

compare said incremented value in said free register to a pre-defined value; and

allocate a spill cell of a spill array corresponding to said index ~~in said free register~~ if said incremented value equals said pre-defined value.

16. (Currently Amended) ~~A~~ The device ~~as in~~ of claim 15, said processor further to determine if said incremented value equals a maximum permitted value.

17. (Currently Amended) An article comprising a storage medium having stored thereon instructions that, when executed by a processor, result in: storing an incremented index of an array in a free register of a processor using a register stack architecture;

calculating in said free register the address of a cell of an array corresponding to said incremented index, said cell storing a value equal to said index;

loading in said free register an incremented value from said cell of said array;

comparing said incremented value in said free register to a pre-defined value; and

allocating a spill cell of a spill array corresponding to said index ~~in said free register~~ if said incremented value equals said predefined value.

18. (Currently Amended) ~~An~~ The article ~~as in~~ of claim 17, wherein said instructions further result in determining if said incremented value equals a maximum permitted value.

19. (Currently Amended) ~~An~~ The article ~~as in~~ of claim 18, wherein said instructions further result in reducing said incremented value if said incremented value equals a maximum permitted value.

20. (Currently Amended) A system comprising:

a dynamic random access memory storage unit; and

a processor with a register stack architecture capable of allocating a spill cell using only one free register.

21. (Currently Amended) ~~A~~ The system ~~as in~~ of claim 20, said processor to

store in said free register an incremented index of an array;

calculate in said free register the address of a cell of an array corresponding to said incremented index, said cell storing a value equal to said index;

load in said free register an incremented value from said cell of said array;

compare said incremented value in said free register to a pre-defined value; and

allocate said spill cell of a spill array corresponding to said index in ~~said free register~~ if said incremented value equals said pre-defined value.

22. (Currently Amended) ~~A~~ The system ~~as in~~ of claim 20, said processor to determine if said incremented value equals a maximum permitted value.

23. (Currently Amended) A processor to:

store an incremented index in a free register of said processor, such processor using a register stack architecture;

calculate in said free register the address of a cell of a first array corresponding to said incremented index, said cell storing a value equal to said index;

load in said free register an incremented value from said cell of said first array;

compare said incremented value in said free register to a pre-defined value; and

allocate a cell of a second array corresponding to said index ~~in said free register~~ if said incremented value equals said predefined value.

24. (Original) The processor of claim 23, the processor to reduce said incremented index modulo to the number of cells in said first array.

25. (Original) The processor of claim 23, the processor to reduce said incremented value if said incremented value equals a maximum permitted value.